LAB MANUAL

Experiment 1

AIM: Generation of the following discrete signals using MATLAB. (i) unit step (ii) unit impulse (iii) unit ramp (iv) Sinc (v) Gaussian

Objective:

1. To generate all basic signals and plot the same as a function of time.
2. To plot the 2D and 3D signals such as LIDAR and Image

<u>Theory:</u>

There are several elementary or basic signals which are used to model a large number of physical signals which occur in nature. These elementary signals are also called Standard Signals. Some of these signals are described below

i) Unit Impulse Function:

An ideal impulse function is a function that is zero everywhere but at the origin, where it is infinitely high.

$$\delta(n) = 1 \; at \; n = 0$$

ii) Unit Step Function:

The unit step function, u(t) is defined as

$$u(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases}$$

That is, *u* is a function of time *t*, and *u* has a value of zero when time is negative and a value of one when time is positive.

iii) Unit Ramp Function:

A ramp function or ramp signal is a type of standard signal which starts at □ = 0 and increases linearly with time. The unit ramp function has unit slope.

$$r(t) = \begin{cases} 0 & \text{for} \quad t<0 \\ t & \text{for} \quad t\geq 0 \end{cases}$$

r(t) = t.u(t)

iv) Sinc Pulse:

A sinc function is an even function with a unit area. A sinc pulse passes through zero at all positive and negative integers (i.e., t=±1,±2,…), but at time t=0, it reaches its maximum of 1. This is a very desirable property in a pulse, as it helps to avoid inter symbol interference, a major cause of degradation in digital transmission systems. The product of a sinc function and any other signal would also guarantee zero crossings at all positive and negative integers.

The normalized sinc function is commonly defined for $x \neq 0$ by

$$\operatorname{sinc} x = \frac{\sin(\pi x)}{\pi x}$$

v) Gaussian Pulse

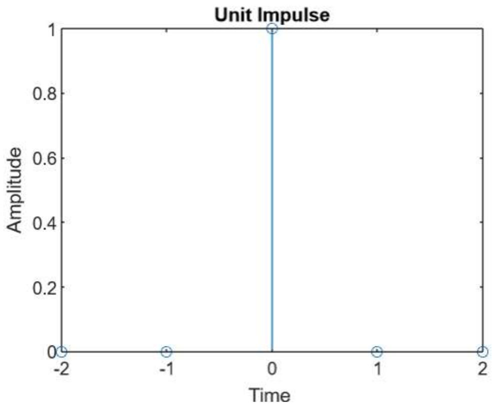
In one dimension, the Gaussian function is the probability density function of the normal distribution,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-(x-\mu)^2/(2\sigma^2)},$$

MATLAB Code:

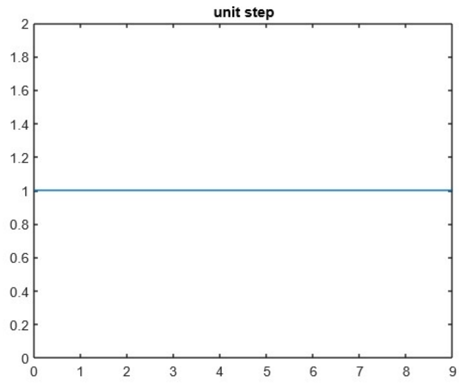
i) Impulse signal

```
%Generation of UNIT impulse signal
clc; close all; clear all;
n=-2:1:2;
y=[zeros(1,2),ones(1,1),zeros(1,2)]
figure(1)
stem(n,y);
xlabel("Time ")
ylabel("Amplitude")
title('unit impulse');
```



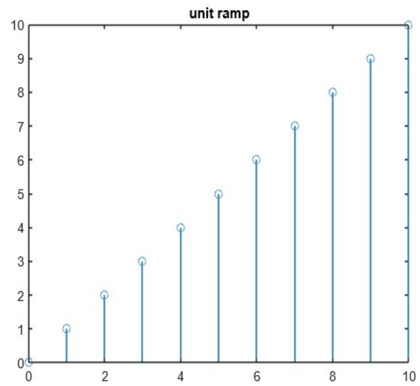
ii) %Generation of UNIT step signal

```
clc; close all; clear all;
n=input('enter the n value');
t=0:1:n-1;
y=ones(1,n);
figure(2)
plot(t,y);
title('unit step');
xlabel("Time (sec)")
ylabel("Amplitude")
```



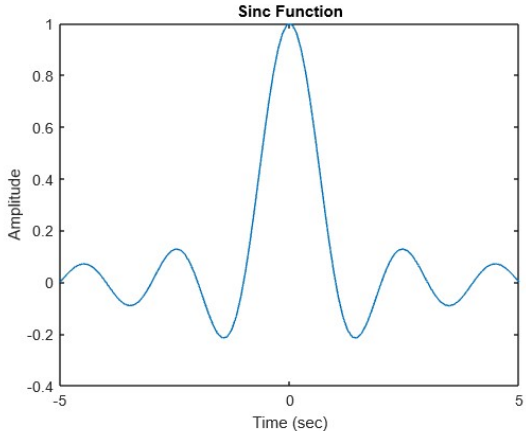
iii)%Generation of unit RAMP signal

```
clc; close all; clear all;
n=input('enter the n value');
t=0:n;
y=t;
figure(3)
stem(y,t);
title('unit ramp');
xlabel("Time (sec)")
ylabel("Amplitude")
```



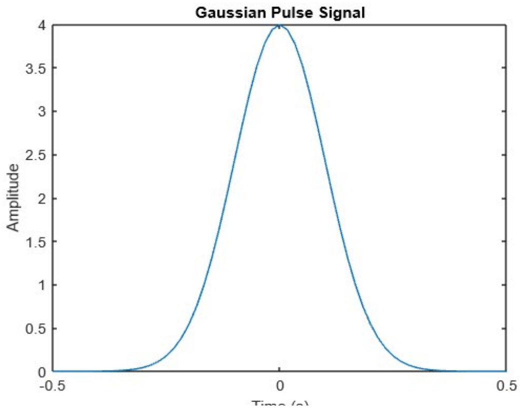
iv) %Generation of sinc pulse

```
t1 = linspace(-5,5);
y1 = sinc(t1);
plot(t1,y1)
xlabel("Time (sec)")
ylabel("Amplitude")
title("Sinc Function")
```



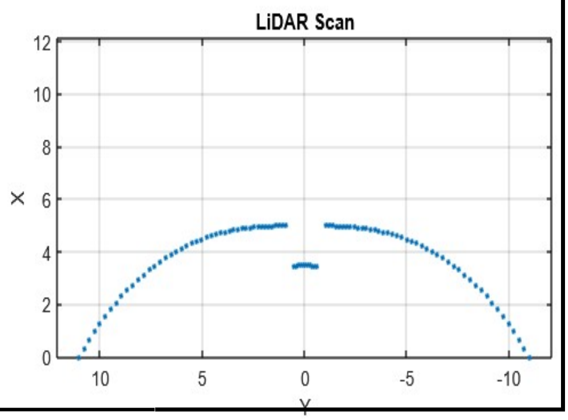
v)%Generation of Gaussian pulse

```
Fs = 60; % sampling freq
t = -.5:1/Fs:.5;
x = 1/(sqrt(2*pi*0.01))*(exp(-t.^2/(2*0.01)));
figure(1);
plot(t,x);
title('Gaussian Pulse Signal');
xlabel('Time (s)');
ylabel('Amplitude');
```

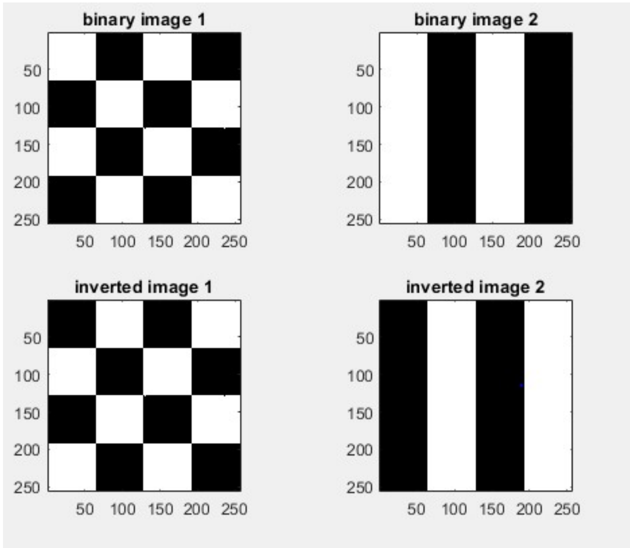


vi) Generation of 2D LIDAR

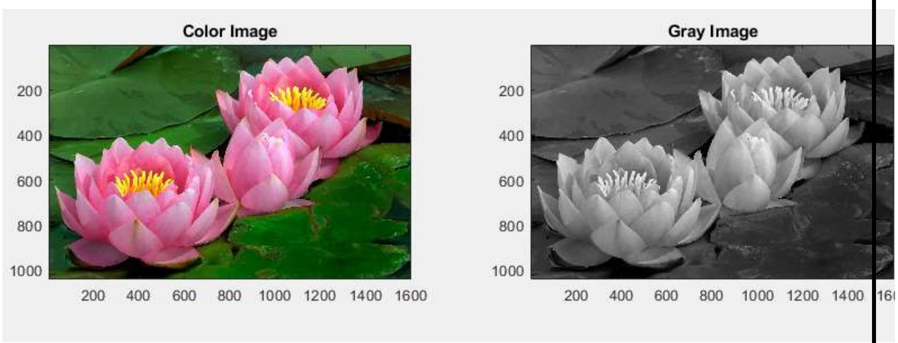
```
x = linspace(-2,2);
ranges = abs((1.5).*x.^2 + 5);
ranges(45:55) = 3.5;
angles = linspace(-pi/2,pi/2,numel(ranges));
scan = lidarScan(ranges,angles);
plot(scan)
```



vii) Generation of 2D Binary image:




```
% create black and white image
clc;% clear command window
clear all;% clear workspace
close all;% clear all figures
w = ones(64,64);
b = zeros(64,64);
bin1= [w b w b
      b w b w
      w b w b
      b w b w];
bin2 = [w b w b
       w b w b
       w b w b
       w b w b];
subplot(2,2,1);subimage(bin1); title('binary image 1');
subplot(2,2,2);subimage(bin2); title('binary image 2');
imwrite(bin1,'bin_image1.tif');
imwrite(bin2,'bin_image2.tif');
i1 = not(bin1);
i2 = not(bin2);
% for block & white image use subimage
subplot(2,2,3);subimage(i1); title('inverted image 1');
subplot(2,2,4);subimage(i2); title('inverted image 2');
```



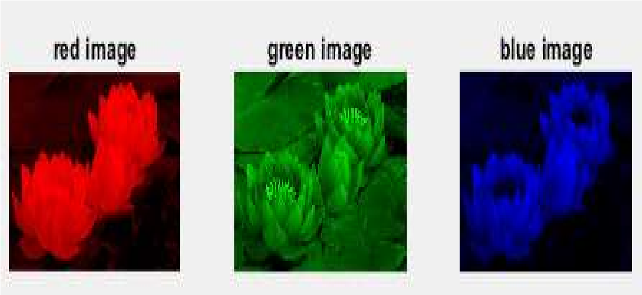

```
%rgb2gray
clc; % clear command window
clear all;% clear workspace
close all;% clear all figures

I1=imread('D:\waterlily.jpg');
figure;
imshow(I1);
title('color image');
I2=rgb2gray(I1);
figure;
imshow(I2);
title('Gray image');
```

```
subplot(2,2,1);subimage(I1); title('Color Image');
subplot(2,2,2);subimage(I2); title('Gray Image');
```

% reading and displaying color image

```
clc;% clear command window
clear all; close all;
a = imread('D:\waterlily.jpg');
[row col dim] = size(a);
figure(1); imshow(a); title('original image');
red = a(:,:,1);% gray scale image of the red plane
green = a(:,:,2);% gray scale image of the green plane
blue = a(:,:,3);% gray scale image of the blue plane
plane = zeros(row,col);
RED = cat(3,red,plane,plane);
GREEN = cat(3,plane,green,plane);
BLUE =  cat(3,plane,plane,blue);
figure(3);
subplot(1,3,1);imshow(RED),title('red image');
subplot(1,3,2);imshow(GREEN),title('green image');
subplot(1,3,3);imshow(BLUE),title('blue image');
```



Practice Questions:

Write a code in Python to generate all the basic signals and plot the same using appropriate library functions.

Sl. No	Criteria	Max Marks	Marks obtained
Data sheet			
A	Problem statement	10	
B	Design & specifications	10	
C	Expected output	10	
Record			
D	Simulation/ Conduction of the experiment	15	
E	Analysis of the result	15	
	Viva	40	
	Total	100	
Scale down to 10 marks			

Experiment 2

AIM: Perform basic operations: time shifting, time scaling and time reversal for the basic signals and plot them as a function of time.

Objective:

1. To perform basic operations on the dependent variable of signals and observe the behavior of signals for different values of amplitude.
2. To perform basic operations on independent variables of signals as a function of time with appropriate shifting and scaling.

Theory:

Basic operations on Signals

The basic set of signal operations can be broadly classified as below.

1. Basic Signal Operations Performed on Dependent Variables

In this transformation, only the quadrature axis values are modified i.e magnitude of the signal changes, with no effects on the horizontal axis values or periodicity of signals like.

- Amplitude scaling of signals
- Addition of signals.
- Multiplication of signals.
- Differentiation of signals.
- Integration of signals.

2. Basic Signal Operations Performed on Independent Variables

- Time Shifting
- Time Scaling
- Time Reversal

Time Shifting

A signal x(t) may be shifted in time by replacing the independent variable *t* by either $t-t_0$ or $t+t_0$. Here t_0 is called the *shifting factor*. Shifting in time may result in time delay or time advancement.

If the independent variable t is replaced by $t-t_0$, the signal is shifted to the right, and the time shift results in a delay of the signal by t_0 units of time. This type of time shifting is known as Right side shifting. This can be achieved by adding t_0 value to every instant in signal x(t).

If the independent variable t is replaced by $t+t_0$, the signal is shifted to the left and the time shift results in an advancement of the signal by t_0 units of time. This type of time shifting is known as Left side shifting. This can be achieved by subtracting t_0 value to every time instant in signal x(t).

Time Scaling

A signal x(t) may be scaled in time by replacing the independent variable *t* with *at*. Here '*a'* is called the *scaling factor*. Time scaling may result in signal compression or signal expansion.

If the independent variable *t* is replaced by *at* and a>1, the signal is *compressed*. This can be achieved by dividing every instant in signal x(t) by '*a*'.

If the independent variable *t* is replaced by *at* and 0<a<1, the signal is *expanded*. This can be achieved by dividing every instant in signal x(t) by '*a*'.

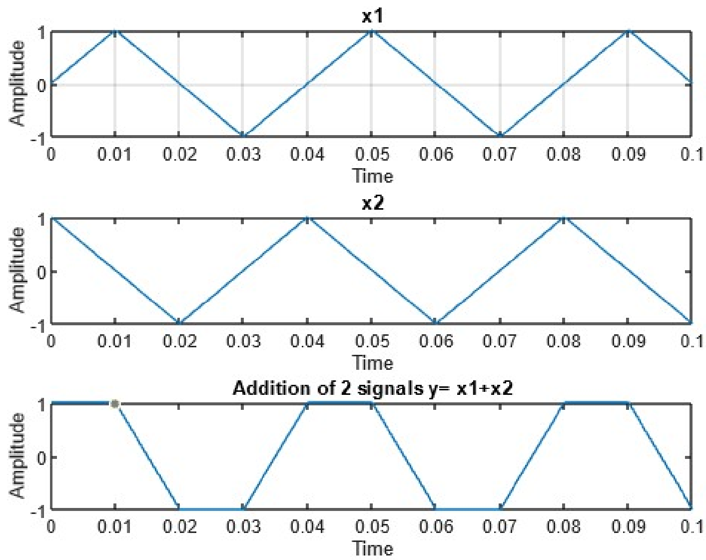
Time Reversal

If the independent variable *t* is replaced by '−*t'* , this operation is known as time reversal of the signal about the y-axis or amplitude axis. This can be achieved by taking a mirror image of the signal x(t) about the y-axis or by rotating x(t) by 180° about the y-axis. Hence, time reversal is known as *folding* or *reflection*.

<u>MATLAB Code:</u>

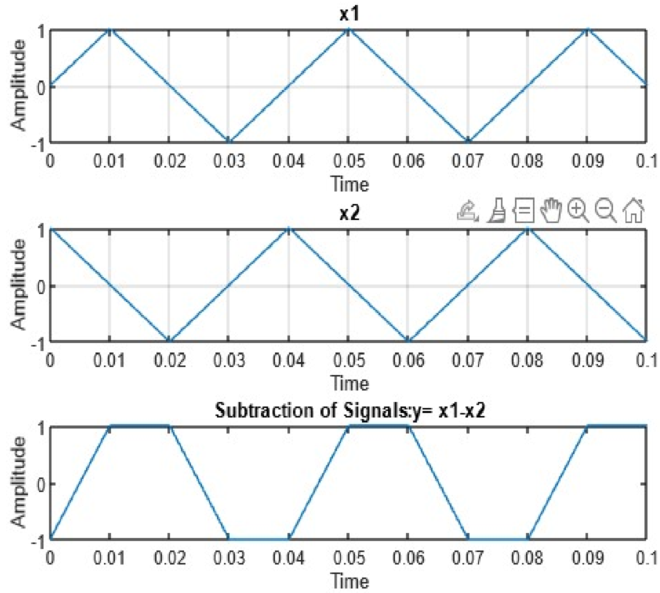
i) Addition of two signals

```
t=0:0.01:0.1
f=25;
t1=2*pi*f*t;
x1=sin(t1);
subplot(3,1,1)
plot(t,x1)
title('x1')
xlabel('Time')
ylabel('Amplitude')
grid on;
x2=cos(t1)
subplot(3,1,2)
plot(t,x2)
title('x2')
```



```
xlabel('Time')
ylabel('Amplitude')
y=x1+x2
subplot(3,1,3)
plot(t,y)
title('Addition of 2 signals y= x1+x2')
xlabel('Time')
ylabel('Amplitude')
```

ii) Subtraction of Signals

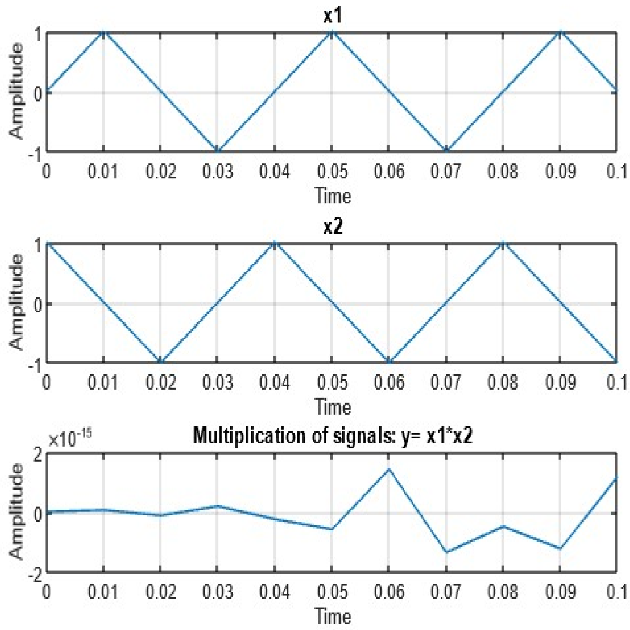



```
t=0:0.01:0.1
f=25;
t1=2*pi*f*t;
x1=sin(t1);
subplot(3,1,1)
plot(t,x1)
title('x1')
xlabel('Time')
ylabel('Amplitude')
grid on;
x2=cos(t1)
subplot(3,1,2)
grid on;
plot(t,x2)
title('x2')
xlabel('Time')
ylabel('Amplitude')
grid on;
y=x1-x2 % Here the subtraction takes place
subplot(3,1,3)
grid on;
plot(t,y)
title('Subtraction of Signals:y= x1-x2')
xlabel('Time')
ylabel('Amplitude')
```

iii) Multiplication of Signals

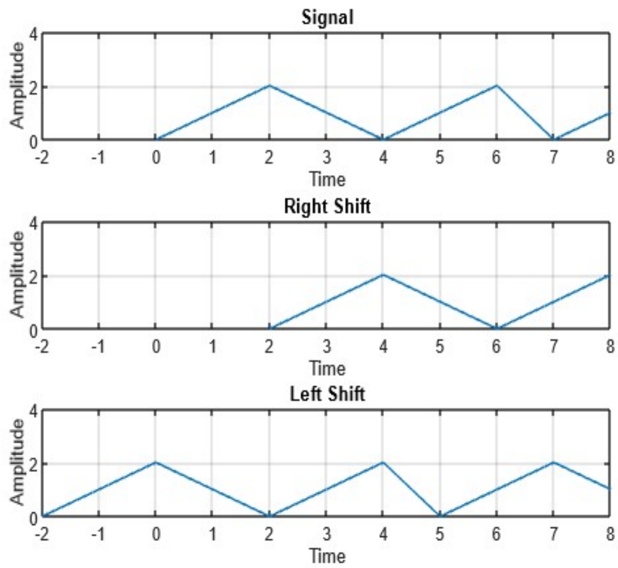
```
t=0:0.01:0.1
f=25;
```

```
t1=2*pi*f*t;
x1=sin(t1);
subplot(3,1,1)
plot(t,x1)
title('x1')
xlabel('Time')
ylabel('Amplitude')
grid on;
x2=cos(t1)
subplot(3,1,2)
plot(t,x2)
title('x2')
xlabel('Time')
ylabel('Amplitude')
grid on;
y=x1.*x2 % Here the multiplication takes place
subplot(3,1,3)
plot(t,y)
title('Multiplication of signals: y= x1*x2')
xlabel('Time')
ylabel('Amplitude')
grid on;
```



i) Time Shifting of Signals

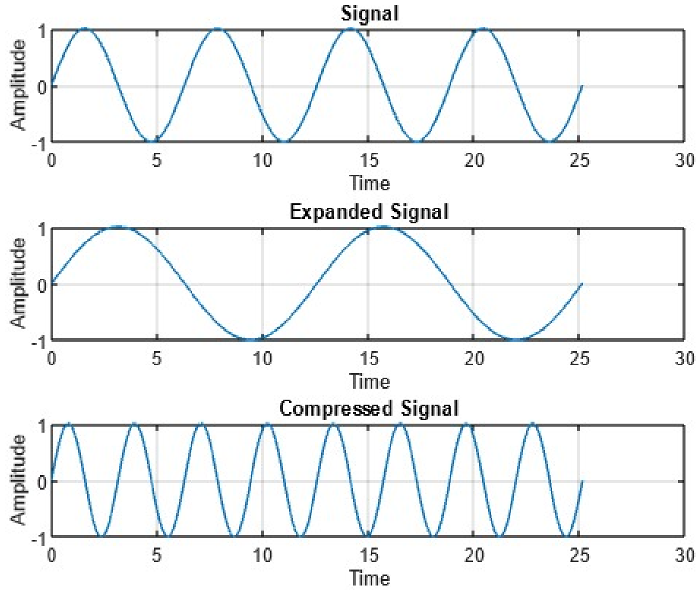
```
t=0:10;
x=[0 1 2 1 0 1 2 0 1 2 1 ];
subplot(3,1,1)
plot(t,x)
title('Signal')
xlabel('Time')
ylabel('Amplitude')
grid on;
axis([-2 8 0 4]);
subplot(3,1,2)
plot(t+2,x)
title('Right Shift')
xlabel('Time')
ylabel('Amplitude')
grid on;
```



```
axis([-2 8 0 4]);
subplot(3,1,3)
plot(t-2,x)
title('Left Shift')
xlabel('Time')
ylabel('Amplitude')
grid on;
axis([-2 8 0 4]);
```

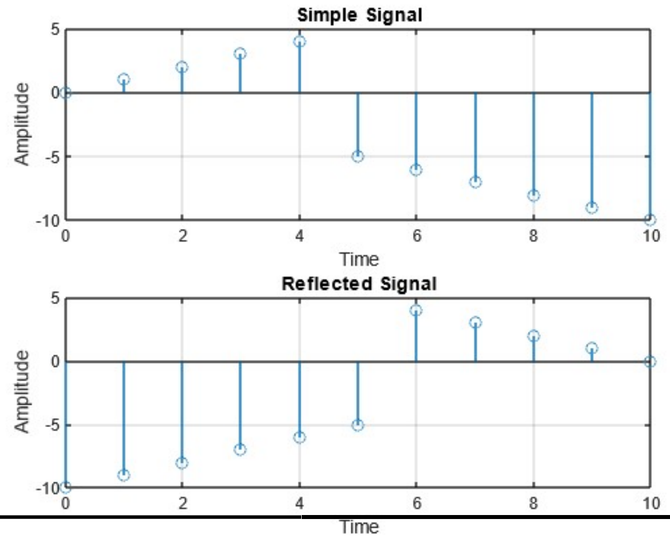
ii)Time Scaling of Signals

```
t=0:0.01:8*pi
x=sin(t);
subplot(3,1,1)
plot(t,x)
title('Signal')
xlabel('Time')
ylabel('Amplitude')
grid on;
y=sin(t/2)
subplot(3,1,2)
plot(t,y)
title('Expanded Signal')
xlabel('Time')
ylabel('Amplitude')
grid on;
z=sin(t*2)
subplot(3,1,3)
plot(t,z)
title('Compressed Signal')
xlabel('Time')
ylabel('Amplitude')
grid on;
```



iii)Time reversal

```
t=0:10;
x=[0 1 2 3 4 -5 -6 -7 -8 -9 -10];
subplot(2,1,1)
stem(t,x)
title('Simple Signal')
xlabel('Time')
```



```
ylabel('Amplitude')
grid on;
y=fliplr(x);
subplot(2,1,2)
stem(t,y)
title('Reflected Signal')
xlabel('Time')
ylabel('Amplitude')
grid on;
```

Practice Questions:

Write the MATLAB Code to sketch the following signals and verify the same using analytical method

a. r(t+2)-r(t+1)-r(t-2)+r(t-3)
b. u(n+2)-3u(n-1)+2u(n-5)

Sl. No	Criteria	Max Marks	Marks obtained
Data sheet			
A	Problem statement	10	
B	Design & specifications	10	
C	Expected output	10	
Record			
D	Simulation/ Conduction of the experiment	15	
E	Analysis of the result	15	
	Viva	40	
	Total	100	
Scale down to 10 marks			

Experiment 3

AIM: To write a MATLAB program to FT of basic signals. Also plot its magnitude and phase spectrum.

Objective:

1. To Generate basic signals Such as sinusoidal, rectangular and sawtooth signals
2. To find their Fourier transform and plot its magnitude and phase spectrum.

Theory:

The generalization of the complex Fourier series is known as the Fourier transform. The term “Fourier transform” can be used in the mathematical function, and it is also used in the representation of the frequency domain. The Fourier transform helps to extend the Fourier series to the non-periodic functions, which helps us to view any functions in terms of the sum of simple sinusoids.

Fourier Transform of a signal x(t) is given by

$$X(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t}dt$$

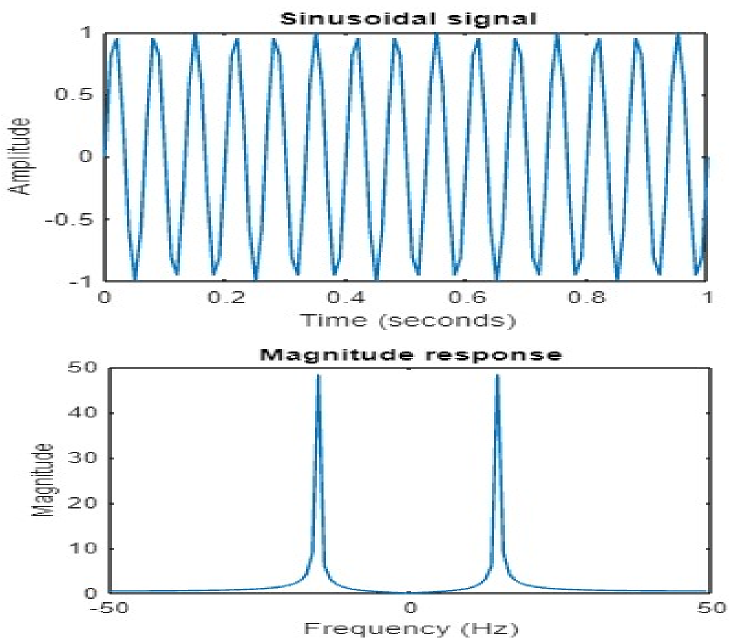
And Inverse Fourier Transform of X(w) is given by

$$x(t) = \sum_{n=-\infty}^{+\infty} X(\omega)e^{j\omega t}\frac{d\omega}{2\pi} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} X(\omega)e^{j\omega t}d\omega$$

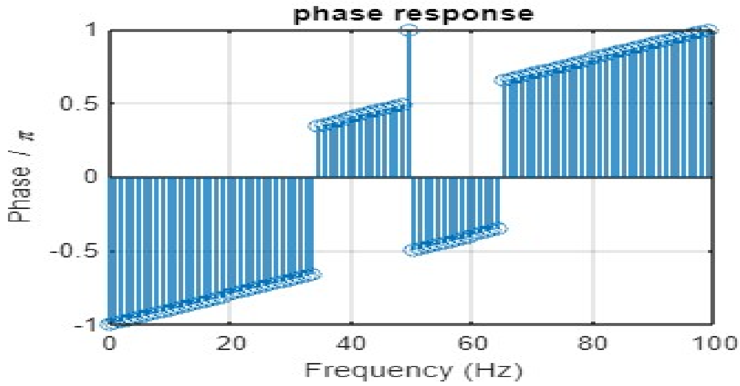
Fourier Transform of Basic Functions

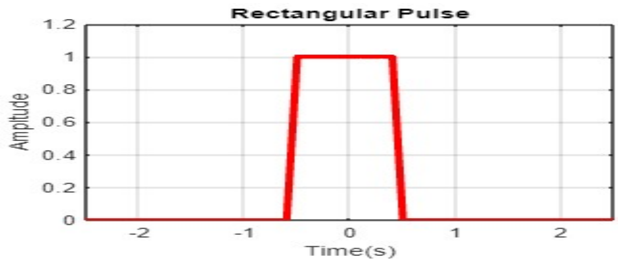
FT of sine signal

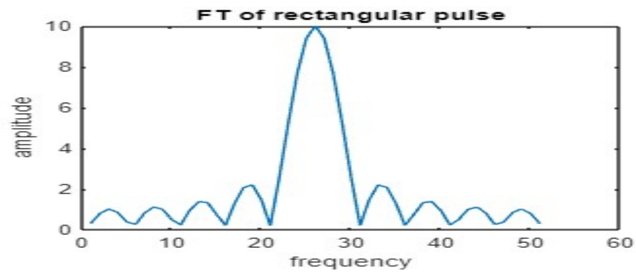
```
%% plotting of signal
Ts=0.01
t = 0:Ts:1
x = sin(2*pi*15*t)
subplot(2,2,1)
plot(t,x)
xlabel('Time (seconds)')
ylabel('Amplitude')
title('Sinusoidal signal')
% Fourier Transform of the signal
y = fft(x);
fs = 1/Ts;
f = (0:length(y)-1)*fs/length(y);
```

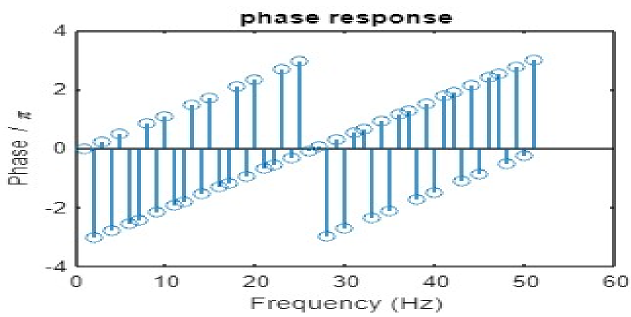


```
n = length(x);
fshift = (-n/2:n/2-1)*(fs/n);
z = fftshift(y);
subplot(2,2,2)
stem(fshift,abs(z))
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('Magnitude response')
theta = angle(z);
subplot(2,2,3)
stem(f,theta/pi)
xlabel("Frequency (Hz)")
ylabel("Phase /\pi")
title('phase response')
%%FT of rectangular pulse
clear all
close all
clc
%fs = 500;
T = 1;
t = -2.5 : 0.1 : 2.5;
x = rectpuls(t,T);
subplot(2,2,1)
plot(t,x,'r','Linewidth',3);
axis([-2.5 2.5 0 1.2])
title({'Rectangular Pulse'})
xlabel({'Time(s)'});
ylabel('Ampltude');
grid
y=fft(x)
subplot(2,2,2)
plot(fftshift(abs(y)))
xlabel('frequency')
ylabel('amplitude')
title('Magnitude response')
theta = angle(y);
subplot(2,2,3)
stem(theta)
xlabel("Frequency (Hz)")
ylabel("Phase / \pi")
```

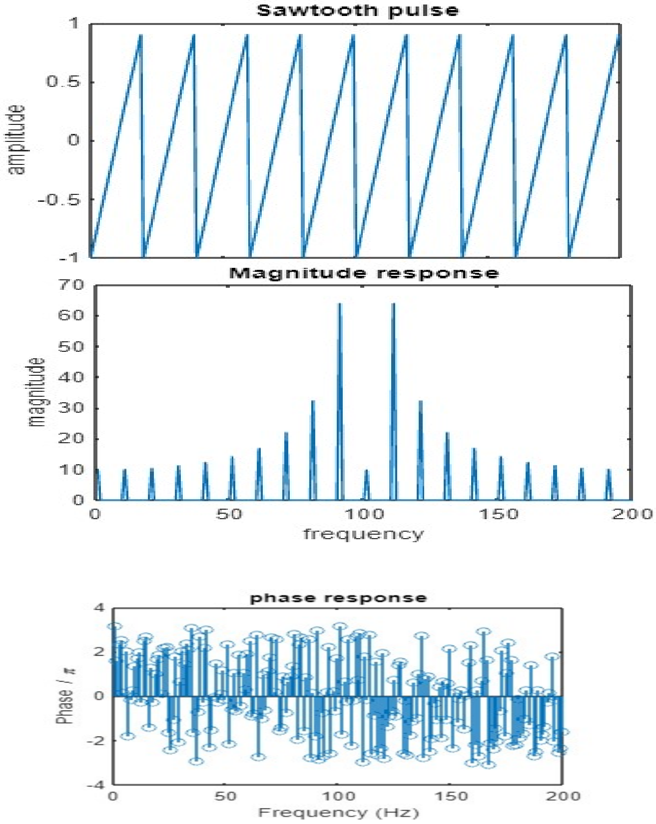








```
title('phase response')
% FT of sawtooth waveform
T = 10*(1/50);
fs = 1000;
t = 0:1/fs:T-1/fs;
x = sawtooth(2*pi*50*t);
subplot(2,2,1)
plot(t,x)
y=fft(x)
subplot(2,2,2)
plot(fftshift(abs(y)))
xlabel('frequency')
ylabel('amplitude')
title('Magnitude response')
theta = angle(y);
subplot(2,2,3)
stem(theta)
xlabel("Frequency (Hz)")
ylabel("Phase / \pi")
title('phase response')
```



Practice Questions:

1. Generate triangular signal and plot its magnitude and phase response
2. Generate an impulse signal (like siren/Hammer/Buzzer)and plot its magnitude and phase response.

Sl. No	Criteria	Max Marks	Marks obtained
Data sheet			
A	Problem statement	10	
B	Design & specifications	10	
C	Expected output	10	
Record			
D	Simulation/ Conduction of the experiment	15	
E	Analysis of the result	15	
F	Viva	40	
	Total	100	
Scale down to 10 marks			

Experiment 4

AIM: To write a MATLAB program for calculating DFT and IDFT discrete time sequences using analytical calculation and inbuilt function.

Objective:

2. To Generate basic signals to find its frequency response..
3. To plot its magnitude and phase spectrum.

Theory:

DFT: Discrete Fourier transform is defined for sequences with finite length.

For a sequence x[n] with length N (x[n] for n=0,1, 2, ..., N-1), the discrete-time Fourier transform is

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} = \sum_{n=0}^{N-1} x[n]e^{-j\omega n}$$

X(w) the discrete-time Fourier transform is periodic with period 2π.

The usually considered frequency interval is (-π , π). There are infinitely many points in the interval. If x[n] has N points, we compute N equally spaced ω in the interval (-π , π). That is, we sample using the frequencies.

$$\omega = \omega_k = \frac{2\pi k}{N}, \qquad 0 \le k \le N-1$$

The above equation is known as the N-point DFT Analysis equation.

$$X(k) = X\left(\frac{k2\pi}{N}\right) = \sum_{n=0}^{N-1} x[n]e^{-j\frac{k2\pi}{N}n} \qquad \omega = \frac{k2\pi}{N}, K = 0,1,\ldots N-1$$

Inverse DFT: The DFT values (X(K), 0 ≤ k ≤ N – 1), uniquely define the sequence x[n] through the inverse DFT formula (IDFT) **x(n) = IDFT {X(k)} ,** **0 ≤ N ≤ n – 1**

The above equation is known as the Synthesis equation.

$$x(n) = \text{IDFT}\ \{X(k)\} = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\ w_n^{\ -kn}, \quad 0 \le N \le n-1$$

Matlab Program:

Write a program to find the frequency response of the signal for

(i) n=10 samples

(ii) n=100 (10 samples plus zero padding)

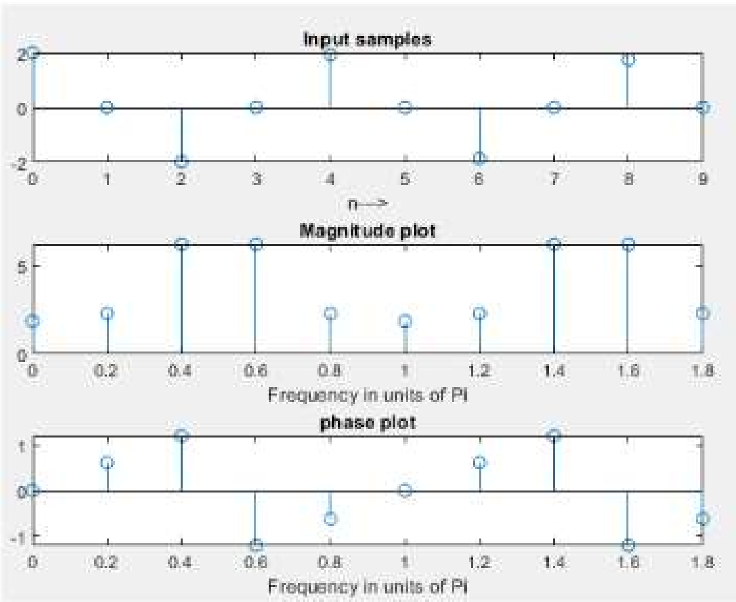
(iii) n=100

Comment on the frequency spectrum.

DFT and IDFT

Matlab code:

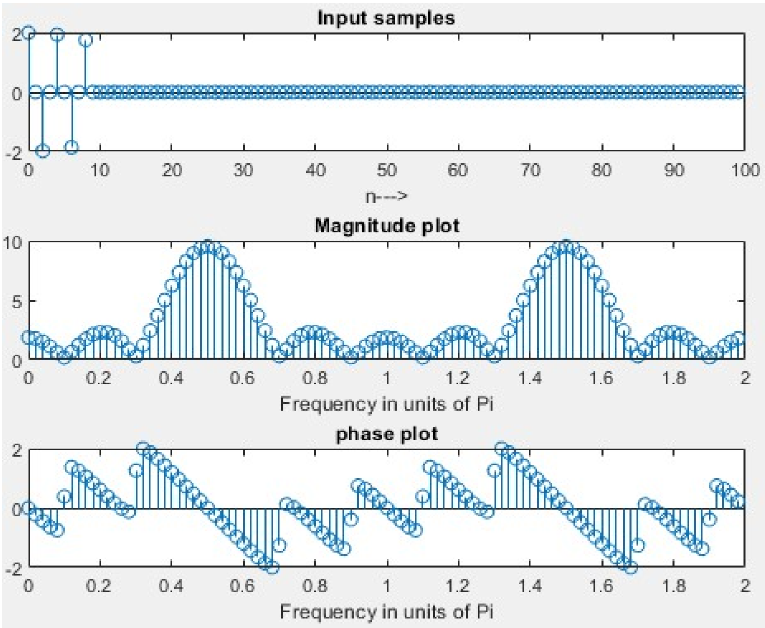
```
% DFT with 10 samples of the signal
n=[0:1:99];
x=cos(0.48*pi*n)+cos(0.52*pi*n);
% Discrete time signal
%   Taking only 10 samples
  n1=[0:1:9];
x1=x(1:1:10);
y1=fft(x1);
mag_y1=abs(y1);
phase_y1=angle(y1);
figure(1);
subplot(3,1,1);
stem(n1,x1);
xlabel('n--->');
title('Input samples');
subplot(3,1,2);
F=2*pi*n1/10;
stem(F/pi,mag_y1);
xlabel('Frequency in units of Pi');
title('Magnitude plot');
X1=ifft(y1);
subplot(3,1,3);
stem(F/pi,phase_y1);
xlabel('Frequency in units of Pi');
title('phase plot');
```



DFT with 10 samples of the signal

% 10 samples + Zero padding

```
n2=[0:1:99];
x2=[x(1:1:10) zeros(1,90)];
y2=fft(x2);
mag_y2=abs(y2);
phase_y2=angle(y2);
figure(2);
subplot(3,1,1);
stem(n2,x2);
xlabel('n--->');
title('Input samples');
subplot(3,1,2);
F=2*pi*n2/100;
stem(F/pi,mag_y2);
xlabel('Frequency in units of Pi');
title('Magnitude plot');
X2=ifft(y2);
subplot(3,1,3);
stem(F/pi,phase_y2);
xlabel('Frequency in units of Pi');
title('phase plot');
```

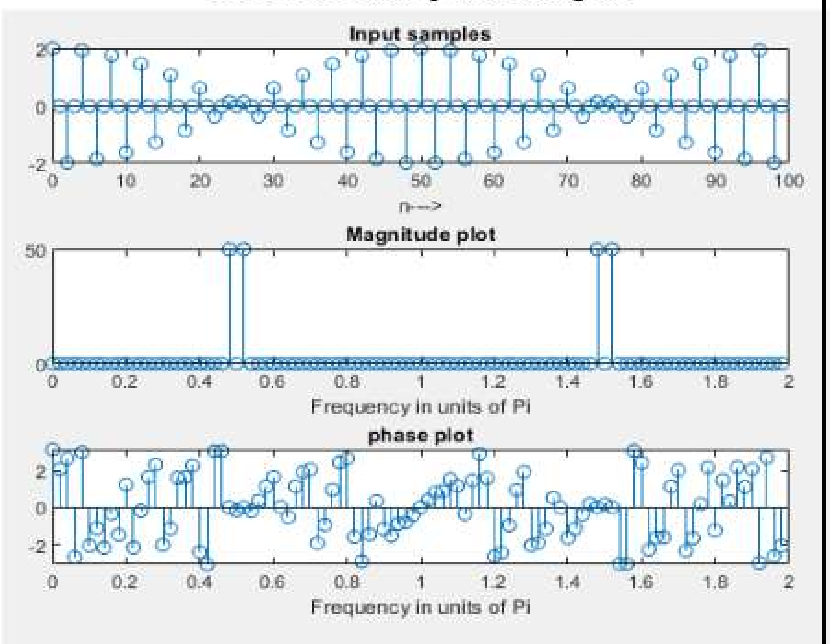


DFT with 10 samples of the signal + padding with zeros

% 100 samples

```
n3=[0:1:99];
x=cos(0.48*pi*n3)+cos(0.52*pi*n3);
x3=x(1:1:100);
y3=fft(x3);
mag_y3=abs(y3);
phase_y3=angle(y3);
figure(3);
subplot(3,1,1);
stem(n3,x3);
xlabel('n--->');
title('Input samples');
subplot(3,1,2);
F=2*pi*n3/100;
%F1=2*pi*[-50:1:49]/100;
stem(F/pi,mag_y3);
xlabel('Frequency in units of Pi');
title('Magnitude plot');
```

DFT with 100 samples of the signal



```
X3=ifft(y3);
subplot(3,1,3);
stem(F/pi,phase_y3);
xlabel('Frequency in units of Pi');
title('phase plot');
```

Practice Questions:

1. Given x(n) = [1,2,3,4] , obtain DFT and IDFT using formula. Plot magnitude and phase plot.
2. Record speech signal and plot FFT.

Sl. No	Criteria	Max Marks	Marks obtained
Data sheet			
A	Problem statement	10	
B	Design & specifications	10	
C	Expected output	10	
Record			
D	Simulation/ Conduction of the experiment	15	
E	Analysis of the result	15	
	Viva	40	
	Total	100	
Scale down to 10 marks			